"SAREE CLASSIFICATION & DETECTION"

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING.

BANGLADESH UNIVERSITY OF BUSINESS & TECHNOLOGY (BUBT)

MIRPUR-2, DHAKA-1216

MARCH, 2021

Saree classification & Detection

A project

Submitted to the Department of Computer Science and Engineering
Bangladesh University of Business and Technology (BUBT), Dhaka
In partial fulfillment of requirements for the full degree of

BACHELOR OF SCIENCE
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted By:
Suborna Nandi(ID: 1415313055)
Rubaiyat Sharmin(ID: 1415313065)
Sharmin Akter(ID: 1415313054)

ABSTRACT

Online cloth business is rapidly growing in Bangladesh and Bangali traditional saree is one of the selected categories of online cloth business. Due to the complex pattern of sarees, it is difficult to distinguish saree from their category. Most common saree categories are katan, jamdani, halfsilk and tangail etc. In this project work, we have developed a machine learning saree classification model using Support Vector Machine (SVM) algorithm. We have collected data from online pages and using google search. After collecting images, we labelled these images into their corresponding categories. We have followed standard machine learning pipelines in our work. We preprocessed the dataset and used Histogram Oriented Gradients (HOG) as image features Due to the large dimension of HOG features, Principal Component Analysis (PCA is used as Dimensionality Reduction technique. We trained PCA on training data and used this PCA model to extract PCA features from testing data also. The SVM model was trained on PCA features. We have collected about 450 images and splitted the dataset into 80% and 20% as training and testing dataset. The training accuracy was 78% and the testing accuracy was 70%. We also have developed a demo application for our project.

ACKNOWLEDGEMENT

"Task successful" makes everyone happy. But the happiness will be gold without glitter if we didn't state the persons who have supported us to make it a success. Success will be crowned to people who made it a reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgment transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped me for the completion of our project work. We consider ourselves lucky enough to get such a good project. This project would add as an asset to our academic profile.

We express our gratitude to the help of our supervisor **Md. Masudul Islam**, for his constant supervision, guidance and co-operation throughout the project and for giving constant motivation and valuable help through the project work. We also would like to thank our honorable chairman **Prof. Dr. Muhammad Firoz Mirdha** for his support and giving us support and giving us permission to use the computer lab whenever we needed.

We extend our sincere gratitude to our parents who have encouraged us with their blessings to do this project successfully. Finally we would like to thank all our friends, all the teaching and non-teaching staff members of the CSE Department, for all the timely help, ideas and encouragement which helped throughout in the completion of the project.

DECLARATION

We hereby declare that the project entitled "**SAREE CLASSIFICATION & DETECTION**" submitted in partial fulfillment by us for the degree B.Sc. Engineering in Computer Science and Engineering in the faculty of Computer Science and Engineering of **Bangladesh University of Business and Technology (BUBT)** under the guidance of our supervision of **Md. Masudul Islam,** Lecturer, department of Computer Science and Engineering is our own work and it contains no material which has been accepted for the award to the candidates of any other disciplines expect few references which is taken from various books and authors to enrich our knowledge about the topic of our project.

-------------------	---------------------	-------------------
Suborna Nandi	Rubaiyat Sharmin	Sharmin Akter
ID:14153103055	ID:14153103065	ID:14153103054
Intake:30	Intake:30	Intake:30

COPYRIGHT

DEDICATION

***Dedicated to our creator all mighty Allah (S.W.T) for his help and our parents for all their love and inspiration*.**

CERTIFICATE

TO WHOM IT MAY CONCERN

That is to certify that Suborna Nandi, Rubaiyat Sharmin and Sharmin Akter students of B.Sc. in CSE has completed their project work titled "**Saree Classification & Detection**" satisfactorily in partial fulfillment for the requirements of B.Sc. in Computer Science and Engineering from Bangladesh University of Business and Technology in the year March, 2021.

--------------------------	---------------------------	------------------------------
Suborna Nandi	Rubaiyat Sharmin	Sharmin Akter
ID:14153103055	ID:14153103065	ID:14153103054
Intake:30	Intake:30	Intake:30

Project Supervisor

MD. MASUDUL ISLAM

Lecturer,
Department of CSE,
Bangladesh University of Business and Technology (BUBT).
Mirpur-2, Dhaka-1216.

APPROVAL

This project "**Saree Classification & Detection**" **Suborna Nandi**, **Rubaiyat Sharmin and Sharmin Akter** respectively our ID No.**14153103055, 14153103065** and **14153103054** students of B.Sc. in Computer Science and Engineering from Bangladesh University of Business and Technology in the year November, 2019 under the supervision of **Md. Masudul Islam,** Lecturer, Department of Computer Science and Engineering has been accepted as satisfactory for the partial fulfillment for the requirements of B.Sc. in Computer Science and Engineering and approved as to its style and contents.

Project Supervisor

Md. Masudul Islam
Lecturer,
Department of CSE,
Bangladesh University of Business and Technology (BUBT).
Mirpur-2, Dhaka-1216.

Chairman

Prof. Dr. Muhammad Firoz Mirdha
Professor and chairman
Department of Computer Science and Engineering（CSE）
Bangladesh University of Business and Technology（BUBT）
Mirpur-2, Dhaka-1216, Bangladesh

Table of Contents

List of Figures

List of Table

CHAPTER 1
INTRODUCTION

1.1 Introduction

Saree is the most demandable cloth item associated with fashion and culture of the womens of Indian subcontinent. Nowadays, fashion is an important part of the economy, including the virtual economy on the Internet [1]. Customers expect online stores to provide them with an easy way to find saree that match their tastes. Therefore, there is a need for high quality clothing search engines. On the other hand, suppliers are not sufficiently prepared to add their products to such search engines because it would require a very accurate, systematic and unified description of their products. Moreover, each store or search engine has a different set of categories and attributes, which is not compatible with others. In order to place a product in a search engine it is necessary to assign it to the appropriate category and apply correct attributes. In the last decades, the problem was solved by manual labeling, and sometimes by constructing classifiers based on manually generated descriptors [2,3,4]. Due to the fact that clothing in online stores is usually well photographed (studio-quality, solid white background), a promising technology for this purpose is deep learning, especially deep machine learning which is proven to be highly successful in classifying images. Image classification refers to the task of extracting information classes from a multiband raster image. The classification process is a multi-step workflow. Therefore, the development of the image classification toolbar is to provide an integrated environment for performing classification using various tools (ArcGIS Desktop). A classifier is needed to distinguish a target object from all the other categories and to make the representations more hierarchical, semantic and informative for visual

recognition. Usually, the Supported Vector Machine (SVM) is the best choice to classify the image [5].

Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. Depending on the interaction between the analyst and the computer during classification, there are two types of classification: supervised and unsupervised. Supervised classification uses the spectral signatures obtained from training samples to classify an image. With the assistance of the Image Classification toolbar, you can easily create training samples to represent the classes you want to extract. You can also easily create a signature file from the training samples, which is then used by the multivariate classification tools to classify the image. Unsupervised classification finds spectral classes (or clusters) in a multiband image without the analyst's intervention. The Image Classification toolbar aids in unsupervised classification by providing access to the tools to create the clusters, capability to analyze the quality of the clusters, and access to classification tools.

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms a core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps: 1) Importing the image via image acquisition tools, 2) Analysing and manipulating the image, 3) Output in which result can be altered image

or report that is based on image analysis. There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital techniques are pre-processing, enhancement, and display, information extraction.

In this project , we have used digital image processing techniques such as gray scale conversion and reshaping each image into a fixed size, width 64 pixels and height 128 pixels. For image classification tasks, we have used the Support Vector Machine (SVM) algorithm.

1.2 Motivation

Saree is the prestigious and cultural part of the Bangladeshi women. Most of the women preferred to wear saree in different social and religious festivals including wedding ceremony, pohela boishakh, birthday party, eid festival, puja festival. In recent years, visual analysis of clothings is a topic that has received increasing attention in computer vision communities. There is already a large body of research on cloth classification and detection based on image. But there are not any systems to classify and detect saree. Considering the above points, we aimed to develop an easily usable model for classifying and detecting the different categories of saree. In this report, we developed a web designed based machine learning model for classification and detection of different kinds of saree. We developed an easily usable model considering all categories of consumers. In this system an image of a saree has to be input and the system will classify the category and components of that saree.

The aim of our project is to make a saree detection system. There are some reasons we motivated to do this project,

i. This system will help to understand the categories of sarees.
ii. The system can be extended for large scale saree detection
iii. The system can be designed and developed for commercial uses
iv. Potential customer can detect and select saree from the system's extended version
v. The system will help a lot of online users to select their desired brand.
vi. This system is profitable.

1.3 Objectives

The main objective of the Saree detection project is to get practical experience on how we can use machine learning and image processing to solve a real life problem. Since, day by day , there is a growing demand for developing intelligent systems to make people's life easier and better, we get inspired to select the project so that we can contribute in the future.

The main objectives of our project are given below,

vii. To get a practical experience of data collection from internet
viii. To learn machine learning project design methodology and technical challenges
ix. To learn image processing, feature engineering and machine learning algorithms
x. To learn server side development of the machine learning model
xi. To learn front end side for developing a demo application for saree detection
xii. To learn python language

1.4 Contributions

The main contribution of our work, probably this is the first attempt to develop such an intelligent solution specially for bangladeshi saree detection using machine learning. There are a lot of challenges in different stages of this project, since a large amount of data collection is costly and time consuming, we have tried to build a prototype application by using all types of requirement software framework and other dependencies.

1.5 Organization of Project Report

The organization of the project report as follows:

In chapter 2, we will discuss literature review where we will mention related research.

In chapter 3, we will discuss system analysis and requirements.

In chapter 4, we will discuss project design.

In chapter 5, we will discuss deployment.

In chapter 5, we will discuss the user manual.

In chapter 6, we add the conclusion part.

1.6 Conclusions

we will discuss. In this chapter, we have described the growing demand of online cloth business, image processing solutions and AI application scope for helping the user to identify bengali cloths. We have mentioned our project objectives and the project report structure.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

A literature review is a survey of scholarly sources (such as books, journal articles, and theses) related to a specific topic or research question. It is often written as part of a thesis, dissertation, or research paper, in order to situate your work in relation to existing knowledge. We have not found so much paper that we can use as reference work. We have found some work on cloth classification and detection. We will mention these works in this chapter.

2.2 Related Works

The authors [6] introduced a complete pipeline for recognizing and classifying people's clothing in natural scenes. This has several interesting applications, including e-commerce, event and activity recognition, on- line advertising, etc. The stages of the pipeline combine a number of state-of-the-art building blocks such as upper body detectors, various feature channels and visual attributes. The core of their method consists of a multi-class learner based on a Random Forest that uses strong discriminative learners as decision nodes. To make the pipeline as automatic as possible they also integrated automatically crawled training data from the web in the learning process. They used 15 clothing classes and introduced a benchmark data set for the clothing classification task consisting of over 80,000 images, which are publicly available.

The authors [7] addressed the data existing dataset size limited problems.They introduced DeepFashion1, a large-scale clothes dataset with comprehensive annotations. It contains over 800,000 images, which are richly annotated with massive attributes, clothing landmarks, and correspondence of images taken under different

scenarios including store, street snapshot, and consumer. Such rich annotations enable the development of powerful algorithms in clothes recognition and facilitating future researches. To demonstrate the advantages of DeepFashion, they proposed a new deep model, namely FashionNet, which learns clothing features by jointly predicting clothing attributes and land- marks. The estimated landmarks are then employed to pool or gate the learned features. It is optimized in an iterative manner. Extensive experiments demonstrate the effective- ness of FashionNet and the usefulness of DeepFashion.

The paper [8] describes a method of clothing classification using a single image. The method assumes to be used for building autonomous systems, with the purpose of recognizing day-to-day clothing thrown casually. A set of Gabor filters is applied to an input image, and then several image features that are invariant to translation, rotation and scale are generated. In this paper, The authors proposed the descriptions of the features with focusing on clothing fabrics, wrinkles and cloth overlaps. Experiments of state description and classification using real clothing show the effectiveness of the proposed method.

The paper [9] considers cloth classification by means of deep neural networks. They redesigned the network structure based on AlexNet, and put forward the deep convolutional neural network model. Experiments are performed on the data sets including ImageNet-1000 and cloth data sets ACS and CAPB. The results show that the proposed deep convolutional neural network is superior to the original AlexNet on these three data sets in terms of accuracy.

2.3 Conclusions

We have mentioned four paper works in this chapter. We found that these works are highly relevant to the fashion and textile industry. The dataset of these mentioned papers are huge in comparison to our collected dataset and they used different machine learning and deep learning methods in their work.

CHAPTER 3
SYSTEM ANALYSIS & REQUIREMENT

3.1 Introduction

System analysis, feasibility study and requirement analysis are essential parts of a standard project development. System analysis helps to identify its goals and purposes and create systems and procedures that will achieve them in an efficient way. Besides, Feasibility study helps to identify the project strong and weak points during the software development life cycle. In the requirement analysis part, we have broadly represented the data requirements and framework requirements.

3.2 System Analysis

System analysis [10] is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. System design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

From our project perspective, the first question is how we can manage the project intelligently. Intelligent part means how a saree can be detected with its category since, a general software or algorithm can't identify an image with its specified object. To overcome this problem, we have addressed the need of using a machine
learning algorithm that can learn image features from data and can distinguish images from different categories.

One of the other technical challenges is to train the machine learning algorithm with enough data. For data collection, we have taken help from many online websites and social media pages.

3.3 Feasibility Study

Feasibility study [11] is an assessment of the practicality of a proposed project or system. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

We have found from our feasibility study that our project is technically feasible. We are able to collect data from online although collecting a lot of data is a challenging task. Since we have a prior knowledge of saree color and category, the data labelling part is quite easy for us. In addition, we have gained knowledge regarding data collection, feature selection and different mining algorithms from our machine learning and data mining course. That is a plus point for us to proceed with the project work.

3.4 Requirement Analysis

Requirements Analysis [12] is the process of defining the expectations of the users for an application that is to be built or modified. It involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore requirements analysis means to analyze, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to facilitate system design.

We have broadly studied our requirements, and we categorize our requirements in two analyses, mainly data requirement and framework requirement for implementation purpose.

3.4.1 Data Requirement

Data is the heart of any machine learning project. Data preparation may be one of the most difficult steps in any machine learning project. The reason is that each dataset is different and highly specific to the project. Nevertheless, there are enough commonalities across predictive modeling projects that we can define a loose sequence of steps and subtasks that you are likely to perform.

This process provides a context in which we can consider the data preparation required for the project, informed both by the definition of the project performed before data preparation and the evaluation of machine learning algorithms performed after.

3.4.2 Framework Requirement

In this project, we have used python language for image processing, model preparation and training. And for demo application, we have used html, css, angular js in front end side.

3.4.2.1 Python Framework Requirement

Python is a comparatively easy language for data processing and machine learning. There are lots of libraries available free to use. We have used some popular libraries. In this below, some libraries are shortly mentioned.

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays [13]. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops, using NumPy. Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array has universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging [14].

Scikit-learn [15] is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible. Scikit-learn integrates well with many other Python libraries, such as matplotlib and plotly for plotting, numpy for array vectorization, pandas dataframes, scipy, and many more [16].

OpenCV [17] is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel. The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations. OpenCV's application areas include, 2D and 3D feature toolkits, Egomotion estimation, Facial recognition system, Gesture recognition, Human–computer interaction (HCI), Mobile robotics, Motion understanding, Object detection, Segmentation and recognition, Stereopsis stereo vision: depth perception from 2 cameras. Structure from motion (SFM), Motion tracking, Augmented reality etc [18].

scikit-image is a Python package dedicated to image processing, and using natively NumPy arrays as image objects. This chapter describes how to use scikit-image on various image processing tasks, and insists on the link with other scientific Python modules such as NumPy and SciPy [19].

Flask [20] is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website. Flask is part of the categories of the micro-framework.

Micro-framework are normally frameworks with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins. In the case of Flask, its dependencies are, Werkzeug a WSGI utility library and jinja2 which is its template engine.

3.4.2.2 Web Framework Requirement

We have used HTML, CSS, ANGULAR.JS for developing the frontend side. These frameworks are described below.

Hypertext Markup Language (HTML) [21] is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Cascading Style Sheets (CSS) [22] is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is

registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

AngularJS [23] is a JavaScript-based open-source front-end web framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. The AngularJS framework works by first reading the Hypertext Markup Language (HTML) page, which has additional custom HTML attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic. The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit Document Object Model (DOM) manipulation with the goal of improving testability and performance.

3.5 Conclusions

In this section, we are discussing system analysis, feasibility study and requirement analysis. After these studies, we have found that there is no potential risk to proceed with study and research to bangali saree detection project.

CHAPTER 4
PROJECT DESIGN

4.1 Introduction

Project design [25] is an early phase of the project where a project's key features, structure, criteria for success, and major deliverables are all planned out. The project design phase might generate a variety of different outputs, including sketches, flowcharts, site trees, HTML screen designs, prototypes, photo impressions and more.

A project design [26] is the first phase of the project cycle. At the beginning, a project develops as an idea or vision-which is feasible. However, the steps to make it feasible is quite difficult. An idea can only become a reality once it is broken down into organized, actionable elements within a timeline.

Some key questions about our project work are ,

1) What is the data source,

2) How we collect the data,

3) Which image processing technique will be used,

4) How we develop the saree detection algorithm

and 5) How we present our system to the audience etc.

The data source is an internet and social media platform. We have collected data by downloading sarees in image format and annotated the image with their corresponding category. Since there are a lot of techniques available for image processing , we have used fixed size and grayscale images for faster processing and for fewer parameter consideration for next processing in the whole pipeline. We have developed the detection system by using a formal classification system.

Formal classification system,

Input Image -> Feature Extraction-> Feature Selection-> Classification Algorithm-> Predicted Category.

We have prepared flowcharts, full system design, data collection workflow etc. In this section, we discuss all these project components.

4.2 System Architecture

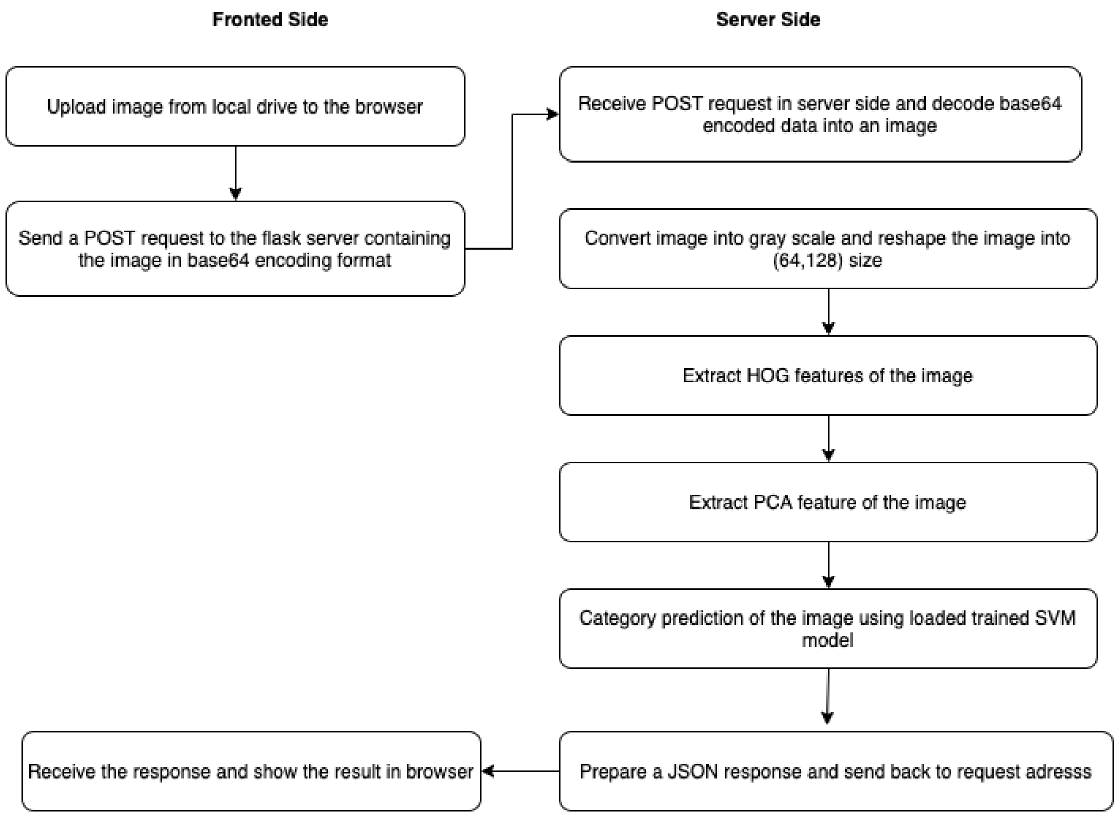


Figure 4.1: Overall project architecture consisting frontend and backend; how an image is sent to the server and processed in the server before returning a json response from the server

The system architecture is the depiction of our demo application for saree detection system. Figure 1 illustrates the process of the frontend and backend part. Users can upload an image from the file directory and when the user sends a request for detecting the saree category, a post request is sent to the backend flask server. Since the pixel data of an image is huge in size, the image data is compressed into base64 encoding and sent with the post request. The flask server is needed to be running in the system. At the server side, the request is received in a specific route function. The post request data is received and data is byte format. The data is decoded from base64 encoding to pixel data. Then the pixel data is loaded into an opencv image object and it is reshaped into (64, 128) format. Since we use only HOG features as the representation of image, the image is converted into grayscale and passed into the HOG feature model. The

output of the HOG feature extraction model is then fed to a PCA model for further feature reduction and important feature selection. The output of PCA model is then fed to the trained SVM model and the result of SVM model processed and sent back to the client side. In the fronted size, the received json response is shown in the browser. This the overall picture of our demo application. More description on Data Collection, Data processing, Feature Extraction, Support Vector Machine, Model training and development are available in the following subsections.

4.3 Data Collection

Data collection [27] is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. Data collection is a research component in all study fields, including physical and social sciences, humanities] and business. While methods vary by discipline, the emphasis on ensuring accurate and honest collection remains the same. The goal for all data collection is to capture quality evidence that allows analysis to lead to the formulation of convincing and credible answers to the questions that have been posed.
A formal data collection process is necessary as it ensures that the data gathered are both defined and accurate. This way, subsequent decisions based on arguments embodied in the findings are made using valid data. The process provides both a baseline from which to measure and in certain cases an indication of what to improve.

We have collected data from different internet sources. We collected images from facebook pages, google image search etc. To maintain quality of the dataset, we check the collected dataset redundancy also. Table 1 shows the total no of collected images. We have collected 166 images for katan, 114 images for jamdani, 73 for tangail and 94 for halfsilk. Due to the complexity of collecting high quality and non-redundant images , we have collected approximately 450 images so far.

Table 4.1 Data Collection for saree classification project

Category	No of collected images
katan	166
jamdani	114
tangail	73
halfsilk	94

4.4 Data Preprocessing

Data processing [28] is the conversion of data into usable and desired form. This conversion or "processing" is carried out using a predefined sequence of operations either manually or automatically. Most of the processing is done by using computers and thus done automatically. The output or "processed" data can be obtained in various forms.

In the data processing part, we converted images into grayscale and use fixed size (64 width, 128 height) for all images.

4.5Feature Extraction

In machine learning, pattern recognition, and image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction [29].

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

Feature extraction involves reducing the number of resources required to describe a large set of data.

When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction [30, 31].

4.5.1 Histogram of Oriented Gradients(HOG)

The histogram of oriented gradients (HOG) [32] is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

Robert K. McConnell of Wayland Research Inc. first described the concepts behind HOG without using the term HOG in a patent application in 1986. In 1994 the concepts were used by Mitsubishi Electric Research Laboratories. However, usage only became widespread in 2005 when Navneet Dalal and Bill Triggs, researchers for the French National Institute for Research in Computer Science and Automation (INRIA), presented their supplementary work on HOG descriptors at the Conference on Computer Vision and Pattern Recognition (CVPR). In this work they focused on pedestrian detection in static images, although since then they expanded their tests to include human detection in videos, as well as to a variety of common animals and vehicles in static imagery.

The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected

regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

a) Gradient computation

The first step of calculation in many feature detectors in image pre-processing is to ensure normalized color and gamma values. As Dalal and Triggs point out, however, this step can be omitted in HOG descriptor computation, as the ensuing descriptor normalization essentially achieves the same result. Image pre-processing thus provides little impact on performance. Instead, the first step of calculation is the computation of the gradient values. The most common method is to apply the 1-D centered, point discrete derivative mask in one or both of the horizontal and vertical directions. Specifically, this method requires filtering the color or intensity data of the image with the following filter kernels.

$$[-1, 0, 1] \text{ and } [-1, 0, 1]^\top .$$

Dalal and Triggs tested other, more complex masks, such as the 3x3 Sobel mask or diagonal masks, but these masks generally performed more poorly in detecting humans in images. They also experimented with Gaussian smoothing before applying the derivative mask, but similarly found that omission of any smoothing performed better in practice.

b) Orientation binning

The second step of calculation is creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is "unsigned" or "signed". Dalal and Triggs found that unsigned gradients used in conjunction with 9 histogram channels performed best in their human detection experiments. As for the vote weight, pixel contribution can either be the gradient magnitude itself, or some function of the magnitude. In tests, the gradient magnitude itself generally produces the best results. Other options for the vote weight could include the square root or square of the gradient magnitude, or some clipped version of the magnitude.

c) Orientation binning

The second step of calculation is creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is "unsigned" or "signed". Dalal and Triggs found that unsigned gradients used in conjunction with 9 histogram channels performed best in their human detection experiments. As for the vote weight, pixel contribution can either be the gradient magnitude itself, or some function of the magnitude. In tests, the gradient magnitude itself generally produces the best results. Other options for the vote weight could include the square root or square of the gradient magnitude, or some clipped version of the magnitude.

d) Descriptor blocks

To account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then the concatenated vector of the components of the normalized cell histograms from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks. R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the

number of channels per cell histogram. In the Dalal and Triggs human detection experiment, the optimal parameters were found to be four 8x8 pixels cells per block (16x16 pixels per block) with 9 histogram channels. Moreover, they found that some minor improvement in performance could be gained by applying a Gaussian spatial window within each block before tabulating histogram votes in order to weight pixels around the edge of the blocks less. The R-HOG blocks appear quite similar to the scale-invariant feature transform (SIFT) descriptors; however, despite their similar formation, R-HOG blocks are computed in dense grids at some single scale without orientation alignment, whereas SIFT descriptors are usually computed at sparse, scale-invariant key image points and are rotated to align orientation. In addition, the R-HOG blocks are used in conjunction to encode spatial form information, while SIFT descriptors are used singly.

Circular HOG blocks (C-HOG) can be found in two variants: those with a single, central cell and those with an angularly divided central cell. In addition, these C-HOG blocks can be described with four parameters: the number of angular and radial bins, the radius of the center bin, and the expansion factor for the radius of additional radial bins. Dalal and Triggs found that the two main variants provided equal performance, and that two radial bins with four angular bins, a center radius of 4 pixels, and an expansion factor of 2 provided the best performance in their experimentation (to achieve a good performance, at last use this configure). Also, Gaussian weighting provided no benefit when used in conjunction with the C-HOG blocks. C-HOG blocks appear similar to shape context descriptors, but differ strongly in that C-HOG blocks contain cells with several orientation channels, while shape contexts only make use of a single edge presence count in their formulation.

e) Block normalization

[to do]

In addition, the scheme L2-hys can be computed by first taking the L2-norm, clipping the result, and then renormalizing. In their experiments, Dalal and Triggs found the L2-hys, L2-norm, and L1-sqrt schemes provide similar performance, while the L1-norm provides slightly less reliable performance; however, all four methods showed very significant improvement over the non-normalized data.

f) Object recognition

HOG descriptors may be used for object recognition by providing them as features to a machine learning algorithm. Dalal and Triggs used HOG descriptors as features in a support vector machine (SVM); however, HOG descriptors are not tied to a specific machine learning algorithm.

https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

4.6 Dimensionality Reduction

Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable.

Feature projection transforms the data from the high-dimensional space to a space of fewer dimensions. The data transformation may be linear, as in principal component analysis (PCA), but many nonlinear dimensionality reduction techniques also exist. For multidimensional data, tensor representation can be used in dimensionality reduction through multilinear subspace learning.

The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance (and sometimes the correlation) matrix of the data is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigenvectors can often be interpreted in terms of the large-scale physical behavior of the system, because they often contribute the vast majority of the system's energy, especially in low-dimensional systems. Still, this must be proven on a case-by-case basis as not all systems exhibit this behavior. The original space (with dimension of the

number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors.

4.6.1 Principal Component Analysis

The principal components of a collection of points in a real p-space that are a sequence of direction vectors, where the i^{th} vector is the direction of a line that best fits the data while being orthogonal to the first *i-1* vectors. Here, a best-fitting line is defined as one that minimizes the average squared distance from the points to the line. These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. [33]

PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data. The i^{th} principal component can be taken as a direction orthogonal to the first *i-1* principal components that maximizes the variance of the projected data.

From either objective, it can be shown that the principal components are eigenvectors of the data's covariance matrix. Thus, the principal components are often computed by eigendecomposition of the data covariance matrix or singular value decomposition of the data matrix. PCA is the simplest of the true eigenvector-based multivariate analyses and is closely related to factor analysis. Factor analysis typically incorporates more domain specific assumptions about the underlying structure and solves eigenvectors of a slightly different matrix. PCA is also related to canonical correlation analysis (CCA). CCA defines coordinate systems that optimally describe the cross-covariance between two datasets while PCA defines a new orthogonal coordinate

system that optimally describes variance in a single dataset. Robust and L1-norm-based variants of standard PCA have also been proposed.

PCA can be thought of as fitting a p-dimensional ellipsoid to the data, where each axis of the ellipsoid represents a principal component. If some axis of the ellipsoid is small, then the variance along that axis is also small.

To find the axes of the ellipsoid, we must first subtract the mean of each variable from the dataset to center the data around the origin. Then, we compute the covariance matrix of the data and calculate the eigenvalues and corresponding eigenvectors of this covariance matrix. Then we must normalize each of the orthogonal eigenvectors to turn them into unit vectors. Once this is done, each of the mutually orthogonal, unit eigenvectors can be interpreted as an axis of the ellipsoid fitted to the data. This choice of basis will transform our covariance matrix into a diagonalised form with the diagonal elements representing the variance of each axis. The proportion of the variance that each eigenvector represents can be calculated by dividing the eigenvalue corresponding to that eigenvector by the sum of all eigenvalues.

4.7 Support Vector Machine

In machine learning, support-vector machines (SVMs, also support-vector networks[1]) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997), SVMs are one of the most robust prediction methods, being based on statistical learning frameworks or VC theory proposed by Vapnik and Chervonenkis (1974) and Vapnik (1982, 1995). Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall [34].

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data is unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The support-vector clustering algorithm, created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support-vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p-1)-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the maximum-margin hyperplane and the linear classifier it defines is known as a maximum-margin classifier; or equivalently, the perceptron of optimal stability.

A support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

4.8 Model training and testing

Data collection is the first step for our work. After data collection we need to preprocess data for model training. Model training means training a model with a dataset so that the model can learn the characteristics of data for the specific classes and internally can separate predictive boundaries for different classes .

Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

Testing a model means to measure the performance of the trained model on unknown data. Model testing is an important step to understand model performance since the objective of model training is to make the model intelligent to understand a group of dataset distinctly from the other group of dataset. If we consider a production environment for machine learning services then we only have the evidence of how the model works on the testing dataset. Testing dataset are only used for model testing , the model doesn’t learn for the dataset.

For model training, we have used the scikit-learn support vector machine library. scikit-learn support vector machines are easy to use and manipulate.

The support vector machine model directly takes the input from the PCA model. The process of image to PCA model is illustrated in the feature extraction part. The dimension of image features after using PCA decreases in number significantly and the parameter space of SVM decreases. Further it also reduces the training time and model convergence complexity respectively. At the training time, we have used cross-validation for better results. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.

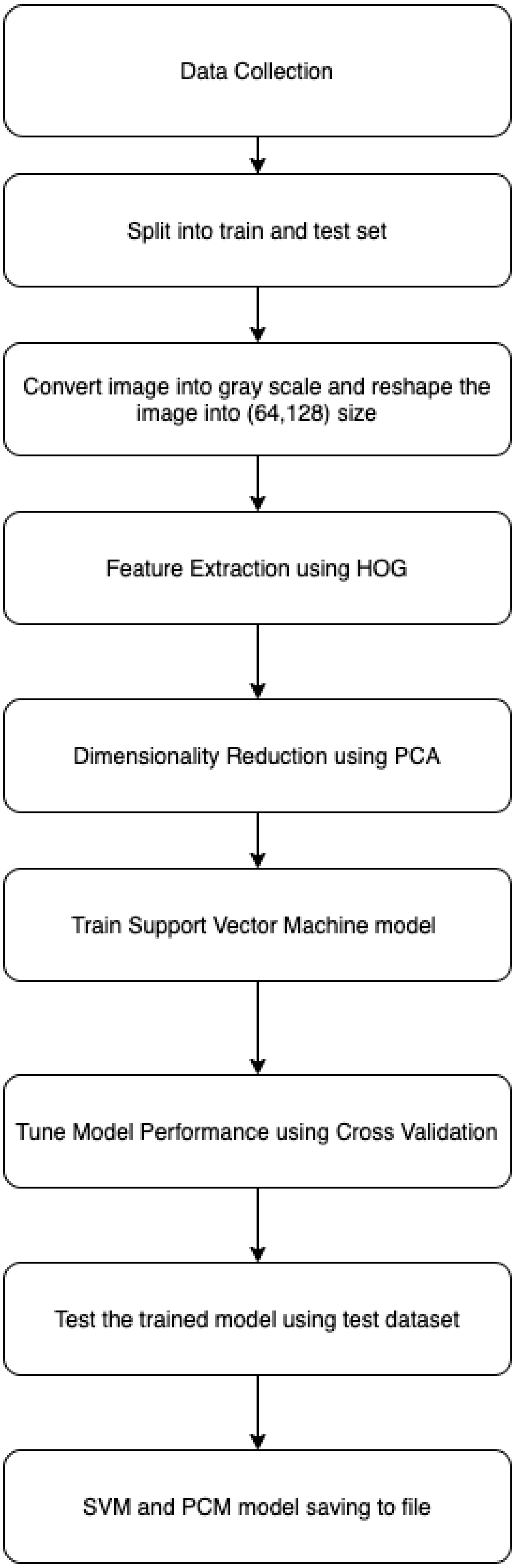


Figure 4. 2: Model training, testing and saving models to the disk

We have used grid search also. The traditional way of performing hyperparameter optimization has been *grid search*, or a *parameter sweep*, which is simply an exhaustive search through a manually specified subset of the hyperparameter space of a learning algorithm. A grid search algorithm must be guided by some performance metric, typically measured by cross-validation on the training set or evaluation on a held-out validation set.

```
# Grid Search
param_grid = [
    {'C': [1, 10, 100, 1000], 'kernel': ['linear']},
    {'C': [1, 10, 100, 1000], 'gamma': [0.001, 0.0001], 'kernel': ['rbf']},
    ]
svc = svm.SVC(probability=True)
clf = GridSearchCV(svc, param_grid)
clf.fit(train_features_pca, train_labels)


print('Training Accuracy:')
y_pred = clf.predict(train_features_pca)
print("Classification report for - \n{}:\n{}\n".format(
    clf, metrics.classification_report(train_labels,y_pred)))


print('Testing Accuracy:')
y_pred = clf.predict(test_features_pca)
print("Classification report for - \n{}:\n{}\n".format(
    clf, metrics.classification_report(test_labels,y_pred)))
```

There are two parameters in this line "clf = GridSearchCV(svc, param_grid)" GridSearchCV function takes two input svc which is regarded as estimator and param_grid which is regarded as parameters

estimator is assumed to implement the scikit-learn estimator interface. Either estimator needs to provide a score function, or scoring must be passed.

param_grid is a dict or list of dictionaries with parameters names (str) as keys and lists of parameter settings to try as values, or a list of such dictionaries, in which case the grids spanned by each dictionary in the list are explored. This enables searching over any sequence of parameter settings. In param_grid we found C, kernel, gamma values. C is a regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty. kernel specifies the kernel type to be used in the algorithm. It must be one of 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed' or a callable. If none is given, 'rbf' will be used. If a callable is given it is used to pre-compute the kernel matrix from data matrices; that

matrix should be an array of shape (n_samples, n_samples). gamman is kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’. If gamma='scale' (default) is passed then it uses 1 / (n_features * X.var()) as value of gamma and if ‘auto’, uses 1 / n_features.

The model is training with this line “clf.fit(train_features_pca, train_labels)” . clf takes the PCA extracted feature data from all training samples with their corresponding labels and grid search based cross validation is applied to get the best fitted support vector machine model which works best for the training dataset. The training accuracy is calculated using the following codes.

```
 print('Training Accuracy:')
 y_pred = clf.predict(train_features_pca)
print("Classification report for - \n{}:\n{}\n".format(
   clf, metrics.classification_report(train_labels,y_pred)))
```

In similar way the testing accuracy is calculated

```
 print('Testing Accuracy:')
y_pred = clf.predict(test_features_pca)
print("Classification report for - \n{}:\n{}\n".format(clf,
metrics.classification_report(test_labels,y_pred)))
```

scikit-learn provides a very useful function for calculating accuracy, sklearn.metrics.classification_report which provides all necessary information like precision, recall, true positive, true negative, false positive and false negative for clear understanding of model performance.

4.9 Model storing

After model training and testing, if we want to use the trained model for further use like model prediction using another python file or model use in the server then we need to store the model information with its parameters. In this work, we have two models for saving, the first one is PCA model and the second one is SVM model. These two models are saved in the file system for further use. We use the following lines of code for saving the model.

```python
# Save PCA model and SVM model
print('Saving model')
with open('pca_model.pickle', 'wb') as handle:
    pickle.dump(pca, handle, protocol=pickle.HIGHEST_PROTOCOL)

with open('svm_model.pickle', 'wb') as handle:
   pickle.dump(clf, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

4.10 Performance Measure and Evaluation Criteria

In this saree classification work, we use a support vector machine model for dress prediction and this is a classification model. To measure the performance measure of the model on both training and test we use precision, recall, f1-score and accuracy parameters. The detailed description of these measure criteria are given in the following section. First of all, the definition of True Positive (TP), True Negative(TN), False Positive (FP) and False Negative(FN) [35] are given:

True Positive (TP): A true positive is an outcome where the model correctly predicts the positive class

True Negative (TN): A true negative is an outcome where the model correctly predicts the negative class.

False Positive (TP): A false positive is an outcome where the model incorrectly predicts the positive class.

False Negative (TP): A false negative is an outcome where the model incorrectly predicts the negative class.

Precision: Precision [36] attempts to answer the following question: What proportion of positive identifications was actually correct?

Precision is defined as follows:

Precision= $\frac{TP}{TP+FP}$

Recall : Recall [36] attempts to answer the following question: What proportion of actual positives was identified correctly?

Mathematically, recall is defined as follows:

Precision= $\frac{TP}{TP+FN}$

Accuracy: Accuracy [37] is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

Accuracy= $\frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

Accuracy= $\frac{TP+TN}{TP+TN+FP+FN}$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

F1-score: F1-score is a measure of a model's accuracy on a dataset. The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. F1-score has the following definition:

F1-score= $\frac{2*(Precision+Recall)}{Precision*Recall}$

4.11 Result Analysis

We have collected approximately 450 pictures of four category sarees and we described the data collection procedure. In this section, we will explain about the experiment analysis.

First of all, we have splitted the dataset into training and testing parts by taking 80% data for training and 20% data for testing. Table2 shows the data description for training and testing cases. We have taken 141 images for katan, 95 images for jamdani, 62 images for tangail and 83 images for halfsilk category for training purpose and for testing purpose we have taken 25,19, 11, 11 images for katan, jamdani, tangail, halfsilk respectively.

Table 4.2 Training and Testing data list

Category	No of training data	No of testing data
katan	141	25
jamdani	95	19
tangail	62	11
halfsilk	83	11

Table2 presents the classification accuracy in both testing and testing data. We have found precision 0.85 and 0.78 in training and testing data respectively. The recall is 0.74 for training data and 0.79 for testing data. We have found f1-score 0.76 and 0.72 on training data and testing data respectively. The testing accuracy was 0.70 where the training accuracy was 0.78. For better clarification, we also compute category wise precision, recall and f1-score.

Table 4.3 Classification accuracy on Training and Testing data

Dataset	Precision	Recall	f1-score	Accuracy
training	0.85	0.74	0.76	0.78
testing	0.78	0.79	0.72	0.70

Table 3 provides category wise precision, recall and f1-score on training data. We have found that the precision for katan and jamdani is 0.70 and 0.78 respectively which is comparatively too less than tangail and halfsilk. The precision for tangail and halfsilk is 0.97 and 0.94 which seems very good. We have also figured out that the recall for katan and jamdani is comparatively better than tangail and halfsilk. The recall of katan, jamdani are 0.94 and 0.80 whereas the recall for tangail and halfsilk are 0.48 and 0.72 respectively. We have found f-score for katan, jamdani, tangail, halfsilk are 0.81, 0.79, 0.65 and 0.82 respectively.

Table 4.4 Category wise precision, recall, f1-score accuracy on Training data

Category	Precision	Recall	f1-score
katan	0.70	0.94	0.81

jamdani	0.78	0.80	0.79
tangail	0.97	0.48	0.65
halfsilk	0.94	0.72	0.82

So far, we have found that katan has low precision 0.70 and tangail has low recall 0.48 among all four categories at training time. Now, we will broadly discuss the category wise testing accuracy.

Table5 shows the category wise performance measure for testing data. The precision scores for katan, jamdani, tangail, halfsilk are 0.65, 0.60. 1.00, 0.88 respectively. Precision scores for tangail, halfsilk are better than katan, jamdani. The similar performance we have found for training data also. For these four categories, the recall values are 0.68, 0.79, 0.64 and 0.64 respectively for katan, jamdani, tangail and halfsilk respectively. The recall value for jamdani is better than among other categories. Now, we have a look on f1-score on testing data. The f1-score are 0.67, 0.68, 0.78, 0.74 for katan, jamdani, tangail and halfsilk categories. The f-score for tangail is better than among other categories.

Table 4.5Category wise precision, recall, f1-score accuracy on Testing data

Category	Precision	Recall	f1-score
katan	0.65	0.68	0.67
jamdani	0.60	0.79	0.68
tangail	1.00	0.64	0.78
halfsilk	0.88	0.64	0.74

4.12 Conclusions

In this chapter, we have discussed system design, project architecture, algorithms, data collection, machine learning training and testing, and result analysis.

CHAPTER 5

DEVELOPMENT

5.1 Introduction

For better user understanding, we have developed a simple webpage to understand the work visually. To connect with the machine learning part, we have deployed the dress classification model into a flask based python web server. The webserver mainly takes the image in a binary data format and returns the category of the data as a json format to the website. In this chapter, we broadly discuss the frontend part and backend part respectively.

5.1 Front End

The frontend part consists of a single webpage. The webpage has an image panel to show an image, a button named choose file to select image from local directory, a classify button to send the image to the backend server and the rest components are two labels for showing the result from the backend. Figure 3 shows the application page.

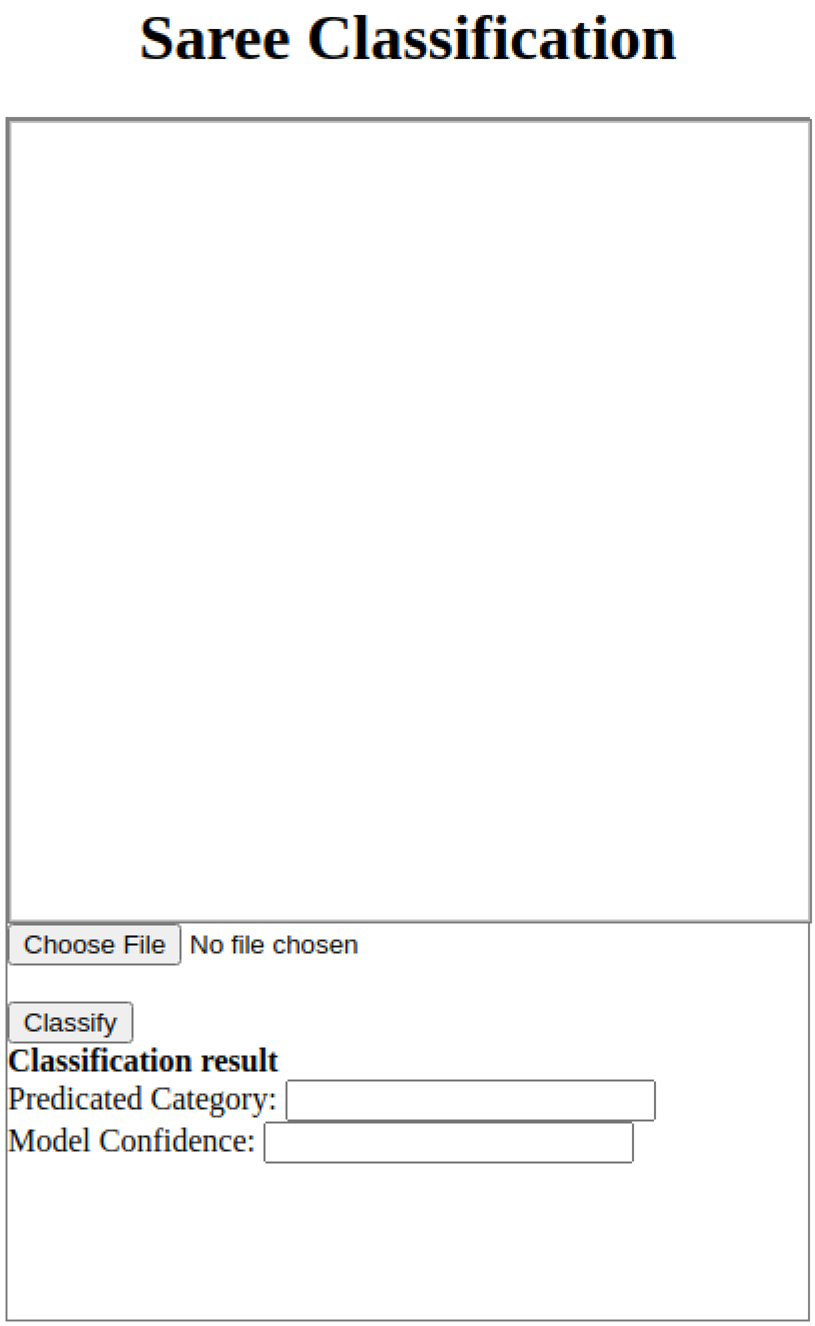


Figure 5.3: Fronted page for saree classification project

After clicking the ‘Choose File’ button and selecting the image from the local directory, the web page looks like Figure 4. The selected image is perfectly adjusted to the image panel. Now we can press a button to send the image to the server.

Figure 5.4: Frontend page view after selecting image from local drive

After sending the image to the local server, we need to wait for about 1 second to get the result. The server will send the output to the page and we found that the resultant data is successfully shown in Figure 3. In Figure 5, we find that the predicted category is katan and the model confidence is about 0.6879. Confidence on all categories are also shown in the page. We find that model confidence for jamdani, katan and tangail are 0.0045, 0.1728 and 0.1348 respectively . The total confidence of all four categories is 1.

Figure 5.5: Frontend page view after receiving response from backend server

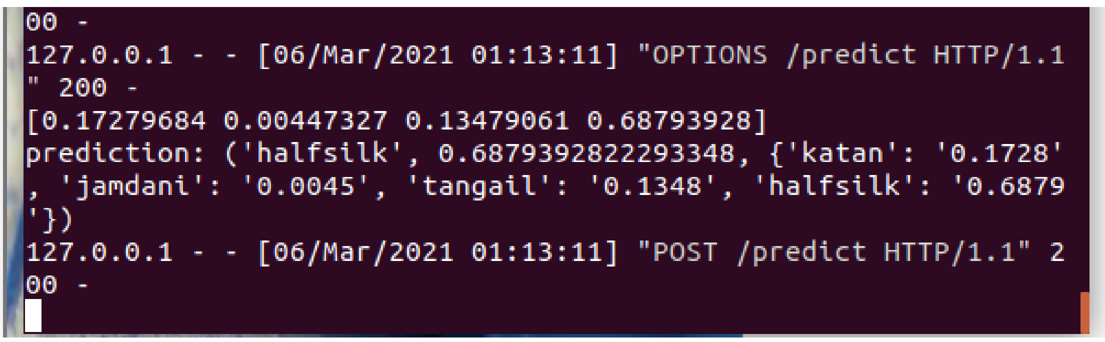

```
00 -
127.0.0.1 - - [06/Mar/2021 01:13:11] "OPTIONS /predict HTTP/1.1
" 200 -
[0.17279684 0.00447327 0.13479061 0.68793928]
prediction: ('halfsilk', 0.6879392822293348, {'katan': '0.1728'
, 'jamdani': '0.0045', 'tangail': '0.1348', 'halfsilk': '0.6879
'})
127.0.0.1 - - [06/Mar/2021 01:13:11] "POST /predict HTTP/1.1" 2
00 -
```

Figure 5.6: Backend server log when processing and predicting the saree category

5.2 Backend

In chapter 2, we have discussed the flask framework. flask is a microweb python framework which helps to deploy the python model. We have defined a route function for accepting input from the http request. The machine model is loaded at one time when the server is started. The preprocessing functions are the same which we have used at model training and testing time. In the part, we have used two already trained PCA and SVM models for feature extraction and category prediction. We have defined a format to send and receive data on the server side. After preprocessing , feature extraction and model prediction, the desired response is sent to the corresponding http user agent. Figure 4 shows the server log after calculating the category prediction result.

5.3 Conclusions

In this deployment part, we primarily want to show how we can do our work in real life applications. We can imagine that we can provide a service for saree category prediction.

CHAPTER 6
USER MANUAL

6.1 Introduction

In this chapter, we will discuss the user manual. A user manual is a technical communication document intended to give assistance to people on how to use a product. A good user manual assists users on how to use a product safely, healthily and effectively.

6.2 System Requirements

System requirement consists of hardware requirement and software requirement. We discuss both in the following subsections.

6.2.1 Hardware Requirements

We have used an average configuration for developing the project. The used PC has 4GB RAM, Corei3 Processor and 1TB HDD. But the project also can be run of lower configuration since it takes low memory and computation lost.

6.2.2 Software Requirements

Python installation is mandatory for this project. We installed Anaconda software to use the conda environment. We use the conda environment to install the project dependencies and to run the project. The python dependencies are numpy, opencv-python, pillow, sklearn, scikit-image, matplotlib. We use a jupyter notebook to write code from the python side. For frontend development we use visual studio code.

6.3 User Interfaces

6.3.1 Create conda environment

The following command is used for creating a conda environment where the conda environment name is dress and python version is 3.

```
conda create -n dress python=3
```

6.3.2 Install project dependencies

To run the project, we need to install the project dependencies in the conda environment. From the terminal, we need to activate the conda environment. Then we can install the project dependencies mentioned in requirements.txt file.

```
conda activate dress
pip install -r requirements.txt
```

6.3.3 Model training

We have prepared a python file named train_hog_pca.py for model training and saving trained models. The file takes two arguments when we want to run the file from the terminal.

```
python train_hog_pca.py argument1 argument2
```

where argument1 = training image folder path, argument2 = testing image folder path

The final command for model training,

```
python train_hog_pca.py data/train/ data/validation/
```

6.3.4 Inference

We have prepared a python file named prediction.py for the trained model inference. The file takes two arguments when we want to run the file from the terminal.

```
python prediction.py argument1 argument2 argument3
```

where argument1=svm model path, argument2=pca model path and argument3=image path

The example command for inference

```
python prediction.py svm_model.pickle pca_model.pickle data/validation/katan/20.jpg
```

6.3.5 Run the server

First of all, we need to run the backend server for showing the demonstration. We can run the server by only typing the following command in the terminal from the project directory. The file internally handles the svm model path and pca model path, so we needn't externally mention it.

```
python dress_classification.py
```

6.3.6 Open the frontend page in browser

The project folder contains a demo.html file. We need to open the file in a browser to demonstrate the frontend work

6.4 Conclusions

In this section, we have described the software and hardware requirements. We also illustrated how we can install the project dependencies for python, training and python server run instructions and finally how we can run fronted.

CHAPTER 7
CONCLUSIONS

7.1 Introduction

Due to the growing online business of saree and clothes in our country, we wanted to work on an AI application project that can be considered as a primary step to detect saree category from image without other people's help. We have faced a lot of difficulties during this project journey. We had to collect data from different online pages and websites and at the data labelling page we had faced a lot of confusions. We have studied how we can use machine learning to develop this system. We have figured out that if we use HOG features rather than colors . shape and other features that will be overall good for getting good accuracy and project work. We have following standard machine learning project pipelines such as data collection, data labelling, image preprocessing, feature extraction, dimensionality reduction, model training and testing, model deployment and finally a fronted page to show the demo. We can ensure that our project can be used as a prototype, if anyone wants to develop an industry level saree classification or cloth classification and detection project.

7.2 Limitation

Although we have finished the project goals in our work still there are some limitations.

a) The amount of collected data is not too large
b) We only considered four popular saree categories

7.3 Future Enhancement

Although we have finished the project goals in our work still there are some limitations.

c) The number of images to individual categories can be increased and the number of categories can be increased from four to five or more.
d) Deep learning approaches can be used instead of machine learning. But we can't ensure it will work better than machine learning since it depends of individual projects
e) The frontend page design can be improved.

References

1. Cychnerski, J., Brzeski, A., Boguszewski, A., Marmolowski, M., & Trojanowicz, M. (2017, September). Clothes detection and classification using convolutional neural networks. In 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA) (pp. 1-8). IEEE.
2. S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-Shop: Cross-Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set," pp. 3330–3337, 2012.
3. M. Yang and K. Yu, "Real-Time Clothing Recognition in Surveillance Videos," in 2011 18th IEEE International Conference on Image Processing, Brussels, 2011.

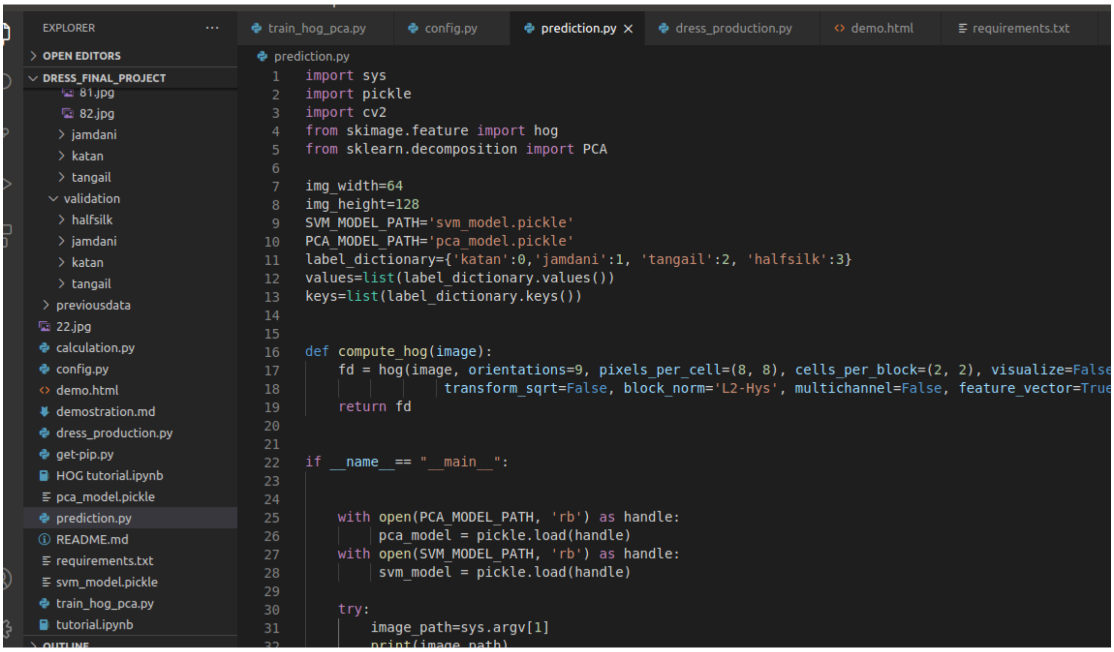
4. H. Chen, A. Gallagher and B. Girod, "Describing Clothing by Semantic Attributes," in Proceedings of the 12th European conference on Computer Vision - Volume Part III, Firenze, 2012.
5. Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.
6. L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. Apparel classification with style. In ACCV, pages 321–335. 2012
7. Z. Liu, P. Luo, S. Qiu, X. Wang, X. Tang, Deepfashion: Powering robust clothes recognition and retrieval with rich annotations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1096–1104.
8. Yamazaki, K., Inaba, M., 2013. Clothing Classification Using Image Features Derived from Clothing Fabrics , Wrinkles and Cloth Overlaps, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 2710-2717. doi:978-1-4673-6357-0/13
9. Fengxin, L., Yueping, L., Xiaofeng, Z.: A novel approach to cloth classification through deep neural networks. In: International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 368–371 (2017)
10. https://en.wikipedia.org/wiki/Systems_analysis
11. https://en.wikipedia.org/wiki/Feasibility_study
12. https://reqtest.com/requirements-blog/requirements-analysis/
13. https://en.wikipedia.org/wiki/NumPy
14. https://numpy.org/
15. https://en.wikipedia.org/wiki/Scikit-learn
16. https://scikit-learn.org/
17. https://en.wikipedia.org/wiki/OpenCV

18. https://pypi.org/project/opencv-python/
19. https://scipy-lectures.org/packages/scikit-image/index.html
20. https://pymbook.readthedocs.io/en/latest/flask.html
21. https://en.wikipedia.org/wiki/HTML
22. https://en.wikipedia.org/wiki/CSS
23. https://en.wikipedia.org/wiki/AngularJS
24. https://machinelearningmastery.com/what-is-data-preparation-in-machine-learning/
25. https://www.wrike.com/project-management-guide/faq/what-is-project-design-in-project-management/
26. https://proposalsforngos.com/what-is-project-design/
27. https://en.wikipedia.org/wiki/Data_collection
28. https://planningtank.com/computer-applications/data-processing
29. https://en.wikipedia.org/wiki/Feature_extraction
30. Sarangi, Susanta; Sahidullah, Md; Saha, Goutam (September 2020). "Optimization of data-driven filterbank for automatic speaker verification". Digital Signal Processing. 104: 102795. arXiv:2007.10729. doi:10.1016/j.dsp.2020.102795. S2CID 220665533.
31. Alpaydin, Ethem (2010). Introduction to Machine Learning. London: The MIT Press. p. 110. ISBN 978-0-262-01243-0. Retrieved 4 February 2012
32. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
33. https://en.wikipedia.org/wiki/Principal_component_analysis
34. https://en.wikipedia.org/wiki/Support-vector_machine

Appendices

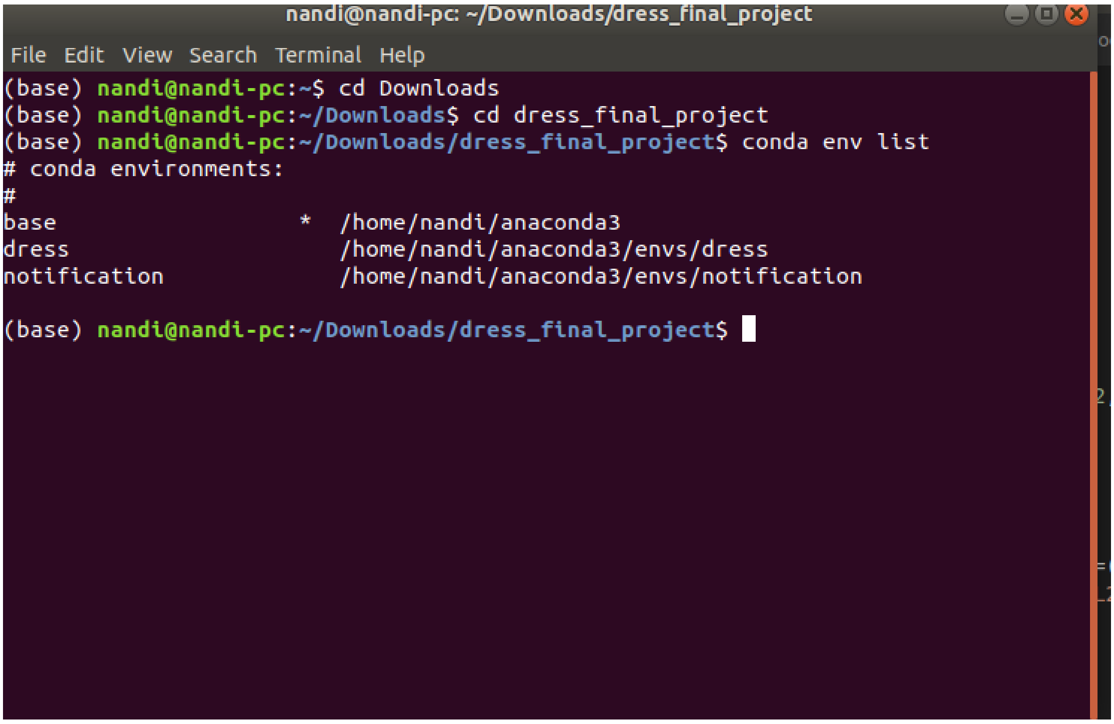
Appendix A

Visual Studio Code:



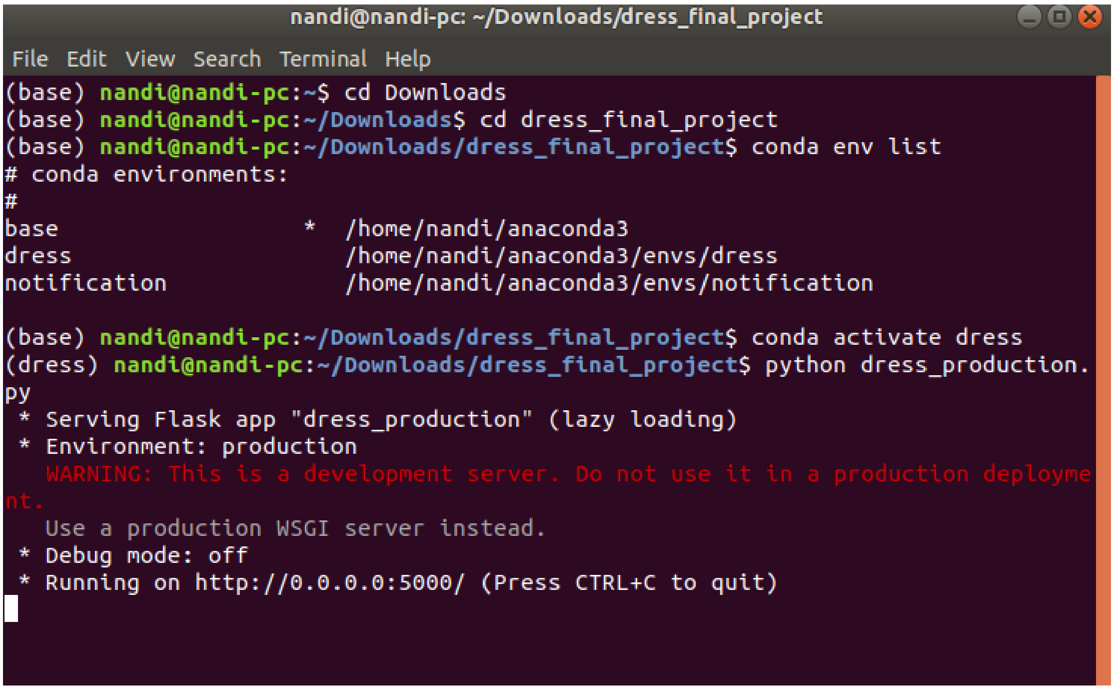
Appendix B

Anaconda:



Appendix C

HTTP Server:




```
(base) nandi@nandi-pc:~$ cd Downloads
(base) nandi@nandi-pc:~/Downloads$ cd dress_final_project
(base) nandi@nandi-pc:~/Downloads/dress_final_project$ conda env list
# conda environments:
#
base                  *  /home/nandi/anaconda3
dress                    /home/nandi/anaconda3/envs/dress
notification             /home/nandi/anaconda3/envs/notification

(base) nandi@nandi-pc:~/Downloads/dress_final_project$ conda activate dress
(dress) nandi@nandi-pc:~/Downloads/dress_final_project$ python dress_production.
py
 * Serving Flask app "dress_production" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```